PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 11/00, 17/50

A2

(11) International Publication Number: WO 99/50746

(43) International Publication Date: 7 October 1999 (07.10.99)

(21) International Application Number: PCT/DK99/00178

(22) International Filing Date: 26 March 1999 (26.03.99)

(30) Priority Data:
0437/98 27 March 1998 (27.03.98) DK

(71) Applicant *(for all designated States except US)*: BAAN VISUALSTATE A/S [DK/DK]; Elkjærvej 30–32, DK–8230 Åbyhøj (DK).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: LEERBERG, Henrik [DK/DK]; Randersvej 180, DK–8200 Århus N. (DK). HULGAARD, Henrik [DK/DK]; Maltagade 3A, 4. th., DK–2300 Copenhagen S. (DK). LIND–NIELSEN, Jørn, Bo [DK/DK]; Øster Allé 25, 3–131, DK–2100 Copenhagen Ø. (DK). ANDERSEN, Henrik, Reif [DK/DK]; Klirevænget 55, DK–2880 Bagsværd (DK). LARSEN, Kim, Guldstrand [DK/DK]; Skovmarken 4, DK–9510 Arden (DK). KRISTOFFERSEN, Kåre, Jelling [DK/DK]; Toldbodgade 21, 4. tv., DK–9000 Aalborg (DK). BEHRMANN, Gerd [DK/DK]; Revlingebakken 36, 1. tv., DK–9000 Aalborg (DK).

(74) Agent: HOFMAN–BANG & BOUTARD, LEHMANN & REE A/S; Hans Bekkevolds Allé 7, DK–2900 Hellerup (DK).

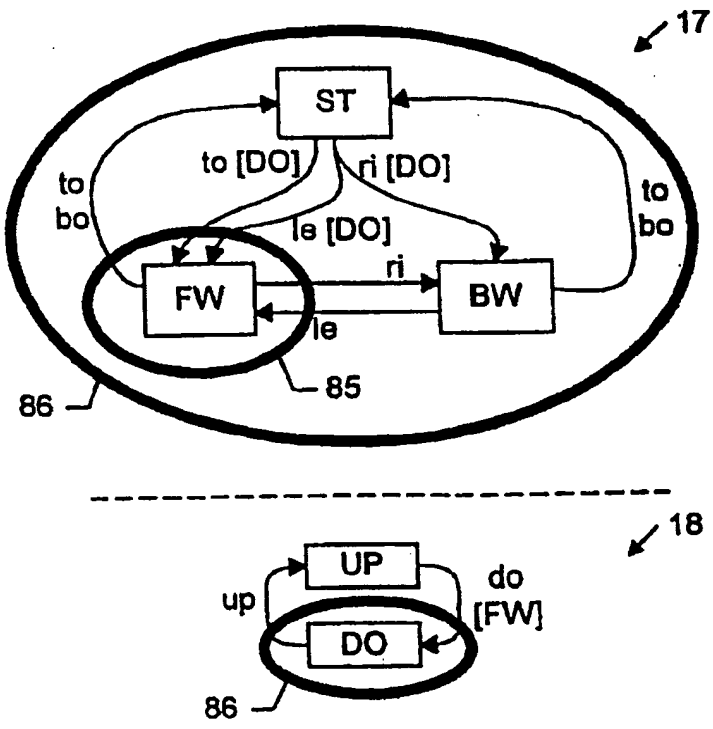
(81) Designated States: AE, AL, AM, AT, AT (Utility model), AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, CZ (Utility model), DE, DE (Utility model), DK, DK (Utility model), EE, EE (Utility model), ES, FI, FI (Utility model), GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SK (Utility model), SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
Without international search report and to be republished upon receipt of that report.

(54) Title: A METHOD AND AN APPARATUS FOR ANALYZING A STATE BASED SYSTEM MODEL

(57) Abstract

The invention relates to a method of analyzing a state based system model comprising a set of machines (M1, .., Mn), said machines each comprising at least one possible state (pS1Mi, .., pSkMi), each machine being in one of its comprised states at any given time, the dynamic behavior of said machines (M1, .., Mn) being defined by predefined transitions between said states of each machine (M1, .., Mn) and dependencies (D) between said machines (Ml, .., Mn). One of many important advantages of the invention is that many analyses of real–life state based system models can be performed without evaluation of a considerable amount of machines in the system model.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

A METHOD AND AN APPARATUS FOR ANALYZING A STATE BASED SYSTEM MODEL

FIELD OF THE ART

The invention relates to a method of analyzing a state based system model comprising a set of machines as stated in the independent claims 1, 11 and 29.

BACKGROUND OF THE INVENTION

When considering the fact that many product segments of the market tend to comprise an increasing amount of embedded software, and as the products within said segments tend to be differentiated more and more only by the performing differences of the embedded software rather than the utilized hardware, the future demands of software designs in general will very great with respect to both fault recognition and elimination and short-term development deadlines.

One of many examples of relevance may be within the automotive industry. Even mass-produced cars tend to comprise an increasing number of dedicated microprocessors. The microprocessors may, e.g., be dedicated to control ABS, fuel injection, light control, different kinds of monitoring, heat control, security systems, etc., and many of the different subsystems will often have to be controlled by a common protocol.

It is evident that the large scale appearance of software-controlled units will cause increasing troubles to the system designers, as it may be difficult to overview every aspect of the possible state of each unit, and of course it may be even more complicated to keep track of the synergy between all the subsystems utilized. A further difficulty which should be mentioned is that most of

the subsystems will be designed by different developers or groups of such, and that the interfaces between such subsystems may be difficult to control as no effective tools can be provided to the developers for the necessary analyses of such large-scale systems together or even in every single unit.

This may cause expensive and crucial delays with respect to the duration of the product development and releasing. It is even more crucial that some products may even be put on the market with inherent hidden errors which under certain unknown conditions, may be triggered and come to light.

This problem is of course really serious in safety critical systems where a defect or fault may even cause injury to persons affected by such a defect.

One way of testing such types of products is to check the logic design prior to the fabrication of a device through symbolic model checking. The technique has turned out to be very efficient for analyses and verification of hardware systems. However, it has not been clear whether model checking is an effective tool for other types of concurrent systems such as e.g. software systems.

One reason why symbolic model checking may not be as efficient is that software systems tend to be both larger and less regularly structured than hardware. For example, many of the results reported for verifying large hardware systems have been for linear structures like stacks or pipelines, for which it is known that the size of the transition relation when represented as a so-called

ROBDD, Reduced Ordered Binary Decision Diagrams, grows linearly with the size of the system.

Another approach to this is described in US patent no. 5,465,216 in which a method of automatic design verification is described. The described method basically accepts the fact that the formal verification suffers from a deficiency of "the state explosion problem", and furthermore concludes that formal verification of very large systems are beyond the capabilities of the current formal verification techniques. Hence, the above-mentioned patent describes a way of decomposing and reducing the system model instead of dealing with the verification method. Consequently, a drawback of the described method is that the possibly obtainable results will only be partial and non-exhaustive.

A more promising technique, based on the above-mentioned ROBDDs, which also exploits the structure of the system is presented in W. Lee et al., Tearing based automatic abstraction for CTL model checking, 1996 IEEE/ACM International Conference on Computer-Aided Design, pages 76-81, San Jose, CA, 1996 IEEE Comput. Soc. Press. This technique uses a partitioned transition relation, and a greedy heuristic is used to select subsets of the transition relation. For each chosen subset, a complete fixed-point iteration is performed. If the formula cannot be proven after this iteration, a larger subset is chosen. In case of an invalid formula the algorithm only terminates when the full transition relation has been constructed (or memory or time has been exhausted). A drawback of the technique is that it uses a greedy strategy involving a fixed-point iteration for each of the remaining machines. If the system only has a single initial

state, as in typical in embedded software systems. The greedy strategy reduces to selecting an arbitrary machine, thus involving extraneous fixed-point iterations.

The present invention meets the requirement of both a formal verification and a use of an unreduced system model and provides the possibility of performing theoretical model "crash tests" in even very large-scale state based system models. Moreover, analyses and verification of the said models can be achieved in non-reduced models at a much higher rate than prior art analyses and verification tools.

SUMMARY OF THE INVENTION

When the method of analyzing a state based system model comprises a set of machines (M1,..,Mn), said machines each comprising at least one possible state (pS1Mi,…,pSkMi) each machine being in one of its comprised states at any given time,

the dynamic behavior of said machines (M1,..,Mn) being defined by predefined transitions between said states of each machine (M1,..,Mn) and dependencies (D) between said machines (M1,..,Mn),

initiating an initial set of at least one machine state (F) of said machines (M1,..,Mn)

initiating a goal set of machine states (A) representing a condition on states of a subset of machines (MI), and repeating the following steps until the analyzing has terminated positively and/or if the subset of machines (MI) comprises all of said machines (M1,..,Mn)

expanding the goal set (A) with a set of states which via transitions can be brought into the previous goal set (A) independently of machines not included in (MI),
if (A) comprises at least one of the set of states in the initial set of states (F) then terminating positively, otherwise expanding the subset of machines (MI) with at least a subset of the machines (M1,..,Mn).

it is possible to obtain a very fast analysis of a state based system model.

Thus, according to the above, the invention deals with a dynamic expansion of a given investigated set of possible states A in a state based system model with the states within the currently investigated machine or set of machines. When the possible states A cannot be expanded any more, i.e. all states in the investigated machines are possible, or when the rest of the states are only possible if certain conditions in other machines is fulfilled, the number of investigated machines is increased and the set of possible states is expanded. This iterative process may continue until certain results are obtained. A desired result could for instance be a verification that a given machine state can be brought into the set of possible initial states A.

Specifically, the invention provides an accurate result when performing so-called reachability check, i.e. when verifying that a set of initial machine states can be brought into certain desired or undesired conditions.

expanding the goal set (A) with a set of states which via transitions can be brought into the previous goal set (A) independently of machines not included in (MI),
if (A) comprises at least one of the set of states in the initial set of states (F) then terminating positively, otherwise expanding the subset of machines (MI) with at least a subset of the machines (M1,..,Mn).

it is possible to obtain a very fast analysis of a state based system model.

Thus, according to the above, the invention deals with a dynamic expansion of a given investigated set of possible states A in a state based system model with the states within the currently investigated machine or set of machines. When the possible states A cannot be expanded any more, i.e. all states in the investigated machines are possible, or when the rest of the states are only possible if certain conditions in other machines is fulfilled, the number of investigated machines is increased and the set of possible states is expanded. This iterative process may continue until certain results are obtained. A desired result could for instance be a verification that a given machine state can be brought into the set of possible initial states A.

Specifically, the invention provides an accurate result when performing so-called reachability check, i.e. when verifying that a set of initial machine states can be brought into certain desired or undesired conditions.

Experimental results have shown that typical so-called reachability checks according to the invention can be performed considerably faster than prior art methods.

Moreover, according to the invention, it is now possible to analyze and verify system models comprising an extremely large number of machines, as no full calculation of all possible global state vectors has to be determined. This important aspect increases the possibility of creating very large state based systems, as a true model of the system can now be established and analyzed completely before marketing of the product, thus eliminating the risk of bringing products on the market with inherent hidden errors.

As many product segments of the market tend to comprise an increasing amount of embedded software, and as the products within said segments tend to be differentiated more and more only by the performing differences of the embedded software rather than the utilized hardware, the future demands of software designs in general will be very great with respect to both the above mentioned fault recognition and elimination and the short-termed development deadlines.

The invention meets the requirement of such a tendency, as the invention provides the possibility of performing theoretical model "crash tests" in even very large-scale state based system models, and moreover analysis and verification of the said models can be achieved in non-reduced models at much higher rate than prior art analyses and verification tools.

It should be noted that the type of test criteria established may vary widely within the scope of the invention. Examples of this could be a verification, an indication of potential dead-lock or, if necessary, a specific and accurate detection of such a dead-lock.

A further advantageous feature of the invention is that the basic compositional structure of step by step expanding the previous search result makes it possible to reuse previously obtained analysis results. Thus, even if a complete investigation of all machines in a system model would become necessary, much will be gained, as no unnecessary calculations will have to be made during the ongoing analyses.

Due to an acknowledgment of a central monotonicity results the previously computed portion of the state space can be reused instead of having to start from scratch each time a new machine is added as in known techniques.

Even when all machines are needed, experiments have shown that the inventive method of including machines one at a time, i.e. exploiting the monotonicity property, is faster than performing a so called traditional fixed-point iteration using a partitioned transition relation and early variable quantification.

In situations where a system model is analysed for reachability with respect to large sequence of goal sets, (A1),.., (An), it may be advantageous for the later reachability problems to reuse already computed results from the earlier problems. In particular, in the backwards expansion from goal set (Ai) early positive termination may be made at any stage, where the cur-

rent expansion of the goal set (Ai) fully contains a previous goal set (Aj) (j<i) for which a positive result has already been obtained.

It should be emphasized that the invention has no restrictions with respect to the way of defining the system "components". The invention is for instance not restricted to simple so-called flat state system models. According to the invention a system may e.g. be defined as a hierarchical state event system comprising hierarchical machines and/or hierarchical states. Bearing this in mind, the invention is preferably advantageously performed in a flat system model, which means that hierarchical systems should preferably be transformed into flat models before an analysis according to the invention is initiated.

According to the invention, states in a system model may, e.g., comprise discrete observations, values of programming variables or registers or latches of a sequential circuit, observations of continuous and time-dependent functions such as temperature, time, speed, altitude, position. Moreover, as mentioned above, states may themselves be system models providing so-called hierarchical system models.

Dependencies are derived from conditions on transitions on other machines in the system model.

Conditions on transitions are either conditions on the current state of other machines in the system model or conditions on the current state of the environment of the system model

Moreover, it should be noted that variations of the expansion criteria or termination conditions may be applicable within the scope of the invention.

When the step of expanding (MI) with at least a subset of the machines (M1,..,Mn) comprises an expansion of (MI) with at least a subset of machines upon which the previous (MI) depends, a very advantageous expansion of MI has been obtained.

The invention performs an analysis of a given system model by incorporating only the machines necessary for the current purpose, i.e. only the machines on which the current evaluated transitions are dependent.

Thus, according to a very preferred embodiment of the invention the expansion of the current investigated MI with machines outside MI should be made, considering that machines without dependencies on the unexpanded set of machines MI would currently provide no further information. Thus, according to the above, preferred embodiment of the invention, the expansion of the investigated set of machines MI is optimized with only the immediately necessary machines. As many analyses of real-life applications can be performed without evaluation of a considerable amount of machines in a system model, an extremely valuable analysis method can be obtained.

It will be appreciated that analyses of large-scale system models will benefit even more from this important feature, as the necessary evaluated space of the system model may be reduced considerably and a great number of evaluations may be avoided.

An important aspect of the above mentioned dynamic ongoing expansion is that a usable result, when analyzing very large scale system models, can only be obtained when considering the dependencies, as described above. Pilot test have in fact shown that almost impossible verifications in prior art systems can now be performed using modest resources on a standard PC.

A further important aspect of the invention is that the difficulties of analyzing reduced state based system models may be eliminated or reduced significantly, as the invention can deal with unreduced system models. It should be noted that the invention may be regarded as a dynamically reduced system model, wherein only the absolutely necessary system model machines are dynamically determined and investigated Thus, the invention benefits from the empirically shown general behavior of state based system models, namely that possible real-life analyses or verifications will only affect a part of all the system model machines.

Basically, it should be noted that transitions between machine states in a given machine are restricted only by the dependencies associated with the specific transitions. Hence, the present method according to the invention requires that a transition without dependencies may be triggered by an event at any given time.

It is evident that, if conditions do in fact exist on the above-mentioned events, they should be incorporated into the system model, if necessary.

The above-mentioned advantageous embodiment of the invention benefits from the structure of a state-based system

model, as it deals with the fact that transitions are basically characterized in two different ways. Some transitions may be fired unconditional, as they may only be dependent on certain known and always possible events, while the other transitions are bound by certain conditions or dependencies to other machines.

Thus, according to the above embodiment, the invention deals with a dynamic expansion of a given investigated set of possible states A in a state based system model with the states within the currently investigated machine or set of machines. When the possible states A cannot be expanded any more, i.e. all states in the investigated machines are possible, or when the rest of the states are only possible if certain conditions in other machines are fulfilled, the number of investigated machines is increased and the set of possible states is expanded. According to the present embodiment it should be noted that the expansion only concerns the machines or some of the machines which have some kind of relevance to the currently investigated machine, i.e. if they have dependencies to the transitions in the currently investigated. This iterative process may continue until certain result is obtained. A desired result could for instance be a verification that a given machine state can be brought into the set of possible states A.

It is evident that the above, preferred embodiment may reduce the duration of the iterations significantly, as the method, so to speak, dynamically neglects the part of the system model which comprises no relevant information for the currently investigated transitions and/or machines.

The present invention provides a technique that significantly improves the performance of e.g. symbolic model checking on embedded reactive systems modeled using a state/event model or other state based models such as state charts.

The invention thus improves the convenience of utilizing state based models, e.g. the control portion of embedded reactive systems, including smaller systems, such as cellular phones, hi-fi equipment, and cruise controls for cars, and large systems, such as train simulators, flight control systems, telephone and communication protocols. The method according to the invention may thus e.g. be used in commercial tools to assist in developing embedded reactive software by allowing the designer to construct a state based model and analyze it by either simulating it or by running a consistency check. The tool automatically generates the code for the hardware of the embedded system. The consistency check is in fact a verification tool that checks for a range of properties that any state based model should have. Some of the checks must be passed for the generated code to be correct, for instance, it is crucial that the model is deterministic. Other checks are issued as warnings that might be design errors such as transitions that can never fire.

State based models can be extremely large. And unlike in traditional model checking, the number of checks is at least linear in the size of the model. The present invention reports results for models with up to 1421 concurrent state machines, and even much larger systems can easily be handled. For systems of this size, traditional symbolic model checking techniques fail, even when using a partitioned transition relation and backward iteration.

The present invention uses a compositional technique that initially considers only a few machines in determining satisfaction of the verification task and, if necessary, gradually increases the number of considered machines. The machines considered may advantageously be determined using a dependency analysis of the structure of the system.

A number of large state based models from industrial applications have been verified, and even the above- mentioned model with 1421 concurrent machines can be verified with modest resources. Compared with known analysis tools the results improve on the efficiency of checking the smaller instances and dramatically increase the size of systems that can be verified.

When the analyzing is terminated negatively after said step of expanding the goal set (A) with a set of states which can be brought into the previous goal set (A) independently of machines not included in (MI) if none of the machines in (MI) are dependent on machines outside (MI), a valid estimate of the system behavior is obtained, as the method according to the invention has been provided, since, when terminated when none of the machines in MI are dependent on machines outside MI, it can be evidently proved that the test criteria cannot be reached. Thus, the designers of even very large scale state based systems have the possibility of forecasting potential run-time problems.

It should be noted that the negative termination, of course, will be absolutely necessary in many applications, as an exact negative indication will often be of

great value. In many types of analyses this negative indication is in fact what the user is looking for. Hence, it will be appreciated that the negative automatic termination itself will be of great importance, and the method of the invention will be far more effective and user-friendly when a kind of automatic termination is incorporated in the method when further iterations are meaningless. It should nevertheless be emphasized that other than this optimal stop criteria can be used.

On the other hand, the invention has the possibility of providing exact knowledge when speaking about positive control of test criteria representing non-desired states or combinations of states. Hence, if the method according to the invention proves that a certain state or a combination of states cannot be obtained under certain conditions, it can evidently be assumed that this situation will not occur in a real-life situation, even though the invention in fact utilizes only a part of the system model during the dynamic test situation. This feature is very important when speaking about a wide spectrum of process applications in which a fault, i.e. a non-desired state, may cause severe damage or confusion.

It should thus be emphasized that a negative as well as a positive termination of the analysis according to the invention may be determined with certainty, which is an extremely useful and valuable feature when performing tests on system models. The invention provides both a very high speed application and accurate and reliable results.

The information following the positive or negative termination can thus be adapted to represent any desired test condition.

Another aspect of the present embodiment of the invention is that optimal termination criteria may be of great importance in a large number of applications, as unnecessary iterations should be avoided, if possible. The above mentioned positive and negative stop criteria ensure that all, but no more than the necessary iterations will be calculated with respect to a reachability analysis. A person skilled in the art will be able to adapt the method of the invention to other desired analysis purposes.

When a visual or audio indication is provided to a user if, after said step of expanding the goal set (A) with a set of states which can be brought into the previous goal set (A) independently of machines not included in (MI), none of the machines in (MI) are dependent on machines outside (MI), a convenient environment of the information provided to a user is obtained.

It should be noted that a user-friendly interface is of even greater importance when a fast interactive process of analyzing can be expected. Not only may the process of analyzing a given system model be accelerated, but the whole design procedure of a state based system model may be shortened considerably.

When the analysis is terminated upon a request from the user, a further advantageous user interface is obtained. Such a kind of interface may e.g. be advantageous when handling large scale systems. Again, as mentioned above, it should be emphasized that the need for a user-friendly interface grows with the capability of the analysis method.

When the dependencies (D) are represented as a directed graph, a further advantageous embodiment is achieved.

A representation as a directed graph, which in itself is a well known data structure for representing dependencies between arbitrary objects, is a very convenient and optimal approach usable for a wide spectrum of analyzing applications.

When the increasing sets of machines (MI) are determined by a breadth-first traversal of the directed graph representing dependencies, a further advantageous embodiment is achieved, as it leads to a minimum dependency closed MI, and thus a fast termination.

This is due to the fact that it includes only the machines on which MI is immediately dependent.

Moreover this method is very efficiently computable.

When the sets of machine states are represented as Reduced Ordered Binary Decision Diagrams (ROBDD's) and the operations upon them are carried out as efficient operations on Reduced Ordered Binary Decision Diagrams (ROBDD's), a further advantageous embodiment according to the invention is obtained.

Thus, efficient operations computing the image of a transition relation on a set of states can be obtained. It should be noted that a computation of the image of a transition relation requires the transition relation to be represented as a single ROBDD, which may sometimes cause problems due to a large size of the representation.

In these cases the transition relation can be more efficiently represented as a disjunction or conjunction of smaller relations called a partitioned transition relation.

When the transitions relation is represented as a partitioned transitions of Reduced Ordered Binary Decision Diagrams (ROBDD's), and the set of states (A) are dynamically computed by an iterative fixed-point iteration, a simple and efficient operation of the invention is obtained implemented by well known techniques in the art.

When the dynamic behaviour of said machines (M1,..,Mn) is defined by predefined transitions between said states of each machine (M1,..,Mn) and dependencies (D) between said machines (M1,..,Mn),

for each machine (M1,..,Mi,..,Mn)

a subset of machines (MI) is initiated to comprise the currently analyzed machine (Mi)

a set (Ai) of living states (Ai) is initiated, said living states being the machine states of the currently analysed machine (Mi) which, independently of other machines, may change state to other possible states (pS1Mi,…,pSkMi) of said machine

the following steps are initiated until the analysis has terminated or if (MI) comprises all machines (M1,..,Mn)

the set of living states (Ai) is expanded with a set of states which via transitions can be brought into the pre-

vious set of living states (Ai) independently of machines not included in (MI)
and/or a set of states which via transitions can be brought to change state of (Mi) independently of machines not included in (MI)
the analysis is terminated positively if (Ai) comprises all possible machine states in said machine (Mi),
otherwise (MI) is expanded with at least a subset of the machines, a further advantageous embodiment is achieved.

As will be understood, the present embodiment of the invention will provide the possibility of detecting all the global states, i.e. a set of machine states, for each machine which may have the possibility to change state under certain possible conditions. Such states will be regarded as living states according to the present terminology.

It is moreover evident that determination of all the living states of each machine may give an indication of possible inherent traps, as a machine state for which a given machine which has no possible transitions to other states may be potential dangerous states, or so-called potential dead states.

It should nevertheless be emphasized that a potential dead state does not necessarily represent an undesired or illegal state. The determined state or states must in fact only be critical if they can be reached from known or given initial system conditions.

When the invention furthermore comprises the steps of each machine (Mi) having potential dead machine states

(Adi) initiating an initial set of machine states (F) of said machine (M1,..,Mn)

initiating a goal set of machine states (Adi) representing the potential dead machine states of machines (MI), and

repeating the following steps until the analyzing has terminated and/or if the subset of machines (MI) comprises all of said machines (M1,..,Mn)

expanding the goal set (Adi) with a set of states which via transitions can be brought into the previous goal set (Adi) independently of machines not included in (MI),
if (Adi) comprises at least one of the states in the initial set of states (F) then terminating positively,
otherwise expanding the subset of machines (MI) with at least a subset of the machines (M1,..,Mn),

further important knowledge about the investigated system model is obtained.

The meaning of a positive termination in the above embodiment of the invention is thus not especially positive, as it has now been determined that the investigated state or combination of potential dead states can actually be reached. A dead-lock has thus been determined, and the machine Mi comprising the determined state or states will not be able to change state, no matter what happens in the surrounding system.

When the method of the invention comprises determining for at least one machine (Mi), at least one of, preferably all, the potential dead machine states (Adi) which,

when said machine (Mi) is in any of said machine states (Adi), independently of possible external events, will remain in the same machine state (Adi),

for each machine (Mi) having potential dead machine states (Adi) initiating an initial set of machine states (F) of said machines (M1,..,Mn)

initiating a goal set of machine states (Adi) representing the potential dead machine states of machines (MI), and

repeating the following steps until the analysis has terminated and/or if the subset of machines (MI) comprises all of said machines (M1,..,Mn)

expanding the goal set (Adi) with a set of states which via transitions can be brought into the previous goal set (Adi) independently of machines not included in (MI),
if (Adi) comprises at least one of the states in the initial set of states (F) then terminating positively, otherwise expanding the subset of machines (MI) with at least a subset of the machines (M1,..,Mn)

a very preferred embodiment of the invention is obtained, as a real dead-lock has been detected.

The invention provides a convincing method of detecting a very unpleasant type of faults, as a deadlock would cause a system, such as a state based system, to enter an endless loop, causing stressing and unreasonable working conditions for the user, at best.

It is evident that detection of reachable dead-locks in a system model provides extremely valuable information to the system designer, as an undetected dead-lock fault may cause severe damage if it should be detected or experienced by a user of a released product. Moreover, it should be emphasized that a fast dead-lock detection, as well as other test criteria, such as the above-mentioned verification analysis, will provide an impressive work tool to a market which is extremely sensitive to release delays and dependent on short term design phases.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by vary examples and not as a limitation in the figures of the accompanying drawings, in which

Figs. 1 - 4 illustrate the basic machines of an embodiment of the invention,

Fig. 5 illustrates the combinations of the machines with the mutual dependencies between the machines of figs. 2-4,

Figs. 6 - 8 illustrate a first example of an embodiment of the invention,

Figs. 9 - 10 illustrate a second example of an embodiment of the invention,

Figs. 11 and 12 illustrate a third example of an embodiment of the invention,